CIS 520: Machine Learning *Due:* Apr. 27, 2021 (11:59 AM EDT)

Problem Set 5

Submission Instructions:

- **PDF Submission:** This is your main submission, and must contain your solutions to **all problems that you are attempting, including both mathematical problems and programming problems.** It must be submitted as a single PDF file to **Gradescope**, compiled in LaTeX using the LaTeX template provided on Canvas. Each problem should be on a new page. Mathematical problems can be either typed in LaTeX or written by hand and **scanned** into images included in your LaTeX solution, but it must be **readable** by our staff to be graded; we recommend that you not take photos of your hand-written solutions. For programming problems, in addition to including any plots, observations, or explanations as requested, be sure to include a snippet of your code in your LaTeX solution; for this, we encourage you to use `lstlisting` environment in `listings` packages for LaTeX.

  Special instruction for submitting on Gradescope: For each problem, select the pages containing your solution for that problem.

- **Code Submission:** You only need to include your code in your PDF submission. There is no separate code submission component.

Summary: Include your code in your PDF submission. The PDF file should be submitted to the corresponding assignment on Gradescope. Plan to submit early!

Collaboration Policy:

You are allowed to discuss problems in groups, but you must write all your solutions and code **individually, on your own**. All collaborators with whom problems were discussed **must** be listed in your PDF submission.

1. (20 points) **Multiclass Perceptron.** Consider an online multiclass classification problem with $K > 2$ classes, where on each trial t, the learner receives an instance $\mathbf{x}_t \in \mathbb{R}^d$, must make a prediction $\widehat{y}_t \in [K]$ (where $[K] = \{1, \dots, K\}$), and then receives the true label $y_t \in [K]$ and incurs zero-one loss $\ell_{0\text{-}1}(y_t, \widehat{y}_t) = \mathbf{1}(y_t \neq \widehat{y}_t)$. Consider the following extension of the perceptron algorithm for this problem, which maintains a weight vector $\mathbf{w}_y^t$ for each class $y \in [K]$, predicts according to the class y whose weight vector $\mathbf{w}_y^t$ has the highest dot product $(\mathbf{w}_y^t)^\top \mathbf{x}_t$ with the current instance $\mathbf{x}_t$, and if a mistake is made, updates only weight vectors $\mathbf{w}_y^t$ corresponding to the true class $y = y_t$ and to classes y that have a higher dot product than the true class, $(\mathbf{w}_y^t)^\top \mathbf{x}_t > (\mathbf{w}_{y_t}^t)^\top \mathbf{x}_t$ (such algorithms are said to be *ultra-conservative*):

```
Algorithm Multiclass Perceptron (MP)
Initial weight vectors $\mathbf{w}_y^1 = \mathbf{0} \in \mathbb{R}^d \ \forall y \in [K]$
For $t = 1, \dots, T$:
  – Receive instance $\mathbf{x}_t \in \mathbb{R}^d$
  – Predict $\widehat{y}_t = \arg\max_{y \in [K]} (\mathbf{w}_y^t)^\top \mathbf{x}_t$
  – Receive true label $y_t \in [K]$
  – Incur loss $\ell_{0\text{-}1}(y_t, \widehat{y}_t)$
  – Update: If $\widehat{y}_t \neq y_t$ then
        $E_t \leftarrow \{ y \in [K] \mid (\mathbf{w}_y^t)^\top \mathbf{x}_t > (\mathbf{w}_{y_t})^\top \mathbf{x}_t \}$
        $\mathbf{w}_y^{t+1} \leftarrow \begin{cases} \mathbf{w}_y^t + \mathbf{x}_t & \text{if } y = y_t \\ \mathbf{w}_y^t - \mathbf{x}_t/|E_t| & \text{if } y \in E_t \\ \mathbf{w}_y^t & \text{otherwise} \end{cases}$
    else
        $\mathbf{w}_y^{t+1} \leftarrow \mathbf{w}_y^t \ \forall y \in [K]$
```

Let $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathbb{R}^d \times [K])^T$, and let $R_2 = \max \{ \|\mathbf{x}_t\|_2 \mid t \in [T] \}$. Suppose there exist some weight vectors $\mathbf{u}_1, \dots, \mathbf{u}_K \in \mathbb{R}^d$ and $\gamma > 0$ such that $\mathbf{u}_{y_t}^\top \mathbf{x}_t - \max_{y \neq y_t} \mathbf{u}_y^\top \mathbf{x}_t \geq \gamma \ \forall t$. Then show that the number of mistakes made by the above algorithm on S satisfies

$$L_S^{0\text{-}1}[\text{MP}] \ \leq \ \frac{2R_2^2 \left(\sum_{y=1}^K \|\mathbf{u}_y\|_2^2 \right)}{\gamma^2} \, .$$

You may find it helpful to note that, by the Cauchy-Schwarz inequality, for any two sets of vectors $\mathbf{a}_y, \mathbf{b}_y \in \mathbb{R}^d$ for $y \in [K]$, $\sum_{y=1}^K \mathbf{a}_y^\top \mathbf{b}_y \leq \sqrt{\left(\sum_{y=1}^K \|\mathbf{a}_y\|_2^2 \right) \left(\sum_{y=1}^K \|\mathbf{b}_y\|_2^2 \right)}$.

2. (25 points) **Semi-Supervised Learning via Naïve Bayes and EM: Hand Simulation.** In this problem you will consider a semi-supervised extension of Naïve Bayes that incorporates unlabeled data via EM. In particular, you will simulate one step of the EM algorithm on a small data set.

For simplicity, consider a binary classification task with 2-dimensional Boolean instances $\mathbf{x} \in \{0, 1\}^2$ and labels $y \in \{\pm 1\}$. Recall that the Naïve Bayes classifier assumes a generative probabilistic model of the form

$$p(\mathbf{x}, y) = p(y) \prod_{j=1}^{2} p(x_j \mid y) \, .$$

In our setting, there are 5 parameters, which we will collectively denote as $\boldsymbol{\theta}$:

$$\begin{aligned} \theta_{+1} &= \mathbf{P}(Y = +1) \, ; \\ \theta_{j|k} &= \mathbf{P}(X_j = 1 \mid Y = k) \, , \quad \text{for each feature } j \in \{1, 2\} \text{ and each label } k \in \{\pm 1\} \, . \end{aligned}$$

Thus the probability of a labeled example $\mathbf{x} = (1,0), y = -1$ under the above model would be

$$\mathbf{P}(Y=-1)\,\mathbf{P}(X_1 = 1 \,|\, Y=-1)\,\mathbf{P}(X_2 = 0 \,|\, Y=-1) = (1-\theta_{+1})\theta_{1|-1}(1-\theta_{2|-1})\,.$$

Standard (Supervised) Naïve Bayes. In the supervised setting, given labeled training data $S_L = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{m_L}, y_{m_L}))$, one computes maximum likelihood estimates as follows:

$$\begin{aligned}
\widehat{\theta}_{+1} &= \frac{1}{m_L}\sum_{i=1}^{m_L} \mathbf{1}(y_i = +1) \\
\widehat{\theta}_{j|k} &= \frac{\sum_{i=1}^{m_L} \mathbf{1}(y_i = k, x_{ij} = 1)}{\sum_{i=1}^{m_L} \mathbf{1}(y_i = k)}
\end{aligned}$$

Given a new instance $\mathbf{x}$, one then uses the estimated parameters together with Bayes rule to compute the probability $P(Y = +1 \,|\, \mathbf{x})$ under the learned model, and classifies the instance as $+1$ if this probability is greater than $\frac{1}{2}$ and as -1 otherwise.

Semi-Supervised Naïve Bayes. In the semi-supervised setting, given labeled training data S_L as above and additional unlabeled data $S_U = (\mathbf{x}_{m_L+1}, \ldots, \mathbf{x}_{m_L+m_U})$, one treats the missing labels in S_U as unobserved variables and uses EM to find maximum likelihood parameter estimates. In particular, one starts with initial parameter estimates $\widehat{\boldsymbol{\theta}}^0$ learned as above from the labeled data S_L alone, and then iteratively estimates posterior distributions on the missing labels of the unlabeled data points (E-step) and updates the parameter estimates via a weighted maximum likelihood estimation (M-step). Specifically, on each round t, in the E-step, for each unlabeled example $\mathbf{x}_i$ one computes the posterior distribution over labels, $q_i^t(k) = P(Y_i = k \,|\, \mathbf{x}_i; \widehat{\boldsymbol{\theta}}^t)$, under the current parameter estimates $\widehat{\boldsymbol{\theta}}^t$; in the M-step, one then updates the parameter estimates as follows:

$$\begin{aligned}
\widehat{\theta}_{+1}^{t+1} &= \frac{1}{m_L + m_U}\left(\sum_{i=1}^{m_L} \mathbf{1}(y_i = +1) + \sum_{i=m_L+1}^{m_L+m_U} q_i^t(+1)\right) \\
\widehat{\theta}_{j|k}^{t+1} &= \frac{\sum_{i=1}^{m_L} \mathbf{1}(y_i = k, x_{ij} = 1) + \sum_{i=m_L+1}^{m_L+m_U} q_i^t(k)\cdot \mathbf{1}(x_{ij} = 1)}{\sum_{i=1}^{m_L} \mathbf{1}(y_i = k) + \sum_{i=m_L+1}^{m_L+m_U} q_i^t(k)}
\end{aligned}$$

On convergence, one uses the final parameter estimates to make predictions on new instances in the same manner as before.

Problem. Suppose you have a training sample of 8 labeled examples and 4 unlabeled examples, distributed as follows (note that since there are only 4 possible instances in our simple setup and 2 possible labels, some examples are repeated in the training sample below; this would be unlikely in real data, but the key ideas present in this example would carry over to real data as well):

Labeled data S_L	**Unlabeled data** S_U
$\mathbf{x} = (x_1, x_2), y$	$\mathbf{x} = (x_1, x_2)$
$(1,1), +1$	$(1,1)$
$(1,1), +1$	$(1,1)$
$(1,0), +1$	$(0,0)$
$(0,0), +1$	$(0,0)$
$(1,0), -1$	
$(0,1), -1$	
$(0,0), -1$	
$(0,0), -1$	
8	4

(a) Calculate the initial maximum likelihood parameter estimates based on the labeled data only: $\widehat{\boldsymbol{\theta}}^0 = (\widehat{\theta}^0_{+1}, \widehat{\theta}^0_{1|+1}, \widehat{\theta}^0_{2|+1}, \widehat{\theta}^0_{1|-1}, \widehat{\theta}^0_{2|-1})$.

(b) For each instance $\mathbf{x} = (x_1, x_2)$ that appears in the unlabeled data, compute the posterior distribution over the label under the parameter estimates computed in the first part above. In particular, consider $\mathbf{x}_9 = \mathbf{x}_{10} = (1,1)$ and $\mathbf{x}_{11} = \mathbf{x}_{12} = (0,0)$ and compute each of the following:

$$\begin{aligned} q^0_9(+1) &= P(Y_9 = +1 \,|\, \mathbf{x}_9 = (1,1); \widehat{\boldsymbol{\theta}}^0) \\ q^0_{11}(+1) &= P(Y_{11} = +1 \,|\, \mathbf{x}_{11} = (0,0); \widehat{\boldsymbol{\theta}}^0) \end{aligned}$$

Show your calculations.

(Hint: Use Bayes' rule.)

(c) Using the results of the first two parts above, find the updated parameter estimates after one step of EM: $\widehat{\boldsymbol{\theta}}^1 = (\widehat{\theta}^1_{+1}, \widehat{\theta}^1_{1|+1}, \widehat{\theta}^1_{2|+1}, \widehat{\theta}^1_{1|-1}, \widehat{\theta}^1_{2|-1})$. Show your calculations.

(d) The log-likelihood of the labeled and unlabeled data $S = (S_L, S_U)$ under parameter estimates $\widehat{\boldsymbol{\theta}}^t$ is given by

$$\ln p(S; \widehat{\boldsymbol{\theta}}^t) = \sum_{i=1}^{m_L} \ln p(\mathbf{x}_i, y_i; \widehat{\boldsymbol{\theta}}^t) + \sum_{i=m_L+1}^{m_L+m_U} \ln \underbrace{\left(\sum_{y_i \in \{\pm 1\}} p(\mathbf{x}_i, y_i; \widehat{\boldsymbol{\theta}}^t) \right)}_{\ln p(\mathbf{x}_i; \widehat{\boldsymbol{\theta}}^t)} .$$

Write an expression for the log-likelihood of the given training data as a function of the 5 parameters $\widehat{\theta}^t_{+1}, \widehat{\theta}^t_{1|+1}, \widehat{\theta}^t_{2|+1}, \widehat{\theta}^t_{1|-1}, \widehat{\theta}^t_{2|-1}$ for any t.

(e) Using the expression derived in the fourth part above, calculate the log-likelihood of the given training data under both the initial parameter estimates $\widehat{\boldsymbol{\theta}}^0$ computed in the first part above and the updated parameter estimates $\widehat{\boldsymbol{\theta}}^1$ computed in the third part above. After one step of EM, has the log-likelihood of the data increased or decreased?

(Hint: You might like to write a small script which given the 5 parameters as input, returns the log-likelihood value, according to the expression you derived in the fourth part above, as output. You can then plug in the values of the parameter estimates obtained above for $t = 0$ and $t = 1$ into your script to obtain the corresponding log-likelihoods. Include a snippet of your code in your LaTeX submission.)

3. (10 points) **Active Learning. Part 1.** Consider a binary classification problem with labels $\{\pm 1\}$. You are given a small labeled training set, from which you learn a logistic regression model. You are also given three more unlabeled data points, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, and are allowed to query the label of one of these. Your logistic regression model predicts the probabilities of each of these instances having label $+1$ as follows:

$$\widehat{\eta}(\mathbf{x}_1) = 0.25\,; \quad \widehat{\eta}(\mathbf{x}_2) = 0.45\,; \quad \widehat{\eta}(\mathbf{x}_3) = 0.75\,.$$

(a) Suppose your final classifier will be evaluated in terms of 0-1 loss. Using an uncertainty sampling approach, which of the above three instances would you choose to query a label for? Briefly explain your answer.

(b) Now suppose your final classifier will be evaluated in terms of the following cost-sensitive loss:

$$\begin{array}{cc|cc} & & \multicolumn{2}{c}{\widehat{y}} \\ & & -1 & +1 \\ \hline y & -1 & 0 & 0.8 \\ & +1 & 0.2 & 0 \end{array}$$

Using an uncertainty sampling approach, which of the above three instances would you choose to query a label for? Briefly explain your answer.

Part 2. Now consider a 3-class classification problem with labels $\{1, 2, 3\}$. You are given a small labeled training set, from which you learn a 3-class logistic regression model. You are also given three more unlabeled data points, $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, and are allowed to query the label of one of these. Your logistic regression model predicts the probabilities of each of these instances belonging to the 3 classes as follows:

$$\begin{pmatrix} \widehat{\eta}_1(\mathbf{x}_1) \\ \widehat{\eta}_2(\mathbf{x}_1) \\ \widehat{\eta}_3(\mathbf{x}_1) \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.4 \\ 0.5 \end{pmatrix} ; \quad \begin{pmatrix} \widehat{\eta}_1(\mathbf{x}_2) \\ \widehat{\eta}_2(\mathbf{x}_2) \\ \widehat{\eta}_3(\mathbf{x}_2) \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.9 \\ 0 \end{pmatrix} ; \quad \begin{pmatrix} \widehat{\eta}_1(\mathbf{x}_3) \\ \widehat{\eta}_2(\mathbf{x}_3) \\ \widehat{\eta}_3(\mathbf{x}_3) \end{pmatrix} = \begin{pmatrix} 0.18 \\ 0.47 \\ 0.35 \end{pmatrix} .$$

(c) Using an uncertainty sampling approach based on *least confident prediction*, which of the above three instances would you choose to query a label for? Briefly explain your answer.

(d) Using an uncertainty sampling approach based on *margin* between the two highest-probability predictions, which of the above three instances would you choose to query a label for? Briefly explain your answer.

4. (20 points) **Markov Decision Processes and Value Iteration.**

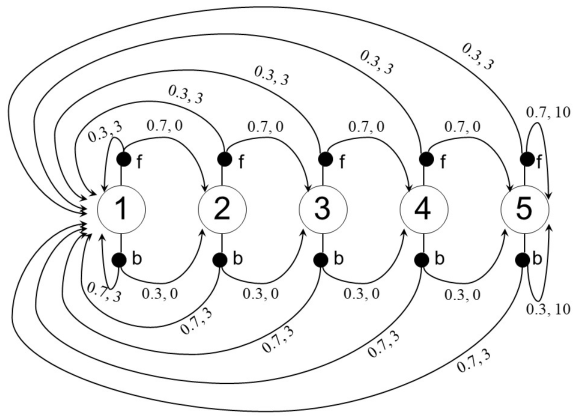

Figure 1: `Chain` environment

In this problem, you will consider the `Chain` environment shown in Figure 1. You will formulate an MDP for this environment and will find an optimal deterministic policy for this environment using value iteration.

Specifically, in this environment, there are 5 states arranged in a chain, and 2 possible actions available to the agent in each state: `f` ("forward") and `b` ("backward"). Once an agent takes an action, there is some stochasticity in how the environment responds, and correspondingly, which state the agent ends up in/what reward it receives. Each action in a state is depicted by a small black circle; the arrows from actions to states depict possible transitions, labeled by the probability with which that transition occurs given the action and previous state, and the associated reward. As can be seen, when the agent takes the `f` action, it usually (with probability 0.7) moves one step forward, and receives zero reward (unless it is in state 5, in which case it usually stays in state 5 and receives a reward of 10), but it sometimes (with probability 0.3) falls back all the way to state 1 and receives a reward of 3. When the agent takes the `b` action, the reverse happens: it usually (with probability 0.7) falls back to state 1 and receives a reward of 3, but sometimes (with probability 0.3) moves a step forward/stays in state 5 and receives a different reward. Assume a discount factor of 0.9.

(a) Formulate an MDP $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ for the above `Chain` environment by specifying each component $\mathcal{S}, \mathcal{A}, p, r$, and γ. For the components p and r, use a tabular format as shown below; specifically, write down $p(s'|s,a)$ and $r(s,a,s')$ for each setting of s, a, s' by filling out the table below (once for p and once for r; you need only fill out entries of the table that are non-zero).

	$a = \mathsf{f}$					$a = \mathsf{b}$				
s	$s' = 1$	$s' = 2$	$s' = 3$	$s' = 4$	$s' = 5$	$s' = 1$	$s' = 2$	$s' = 3$	$s' = 4$	$s' = 5$
1										
2										
3										
4										
5										

(b) Write a small piece of code to implement the value iteration algorithm; your program should take as input a finite MDP, and return as output the optimal value function V^*. In your code, initialize $V^0(s) = 0$ for all s, and stop when $\|V^{t+1} - V^t\|_\infty = \max_s |V^{t+1}(s) - V^t(s)|$ becomes smaller than 0.001. Use your implementation to find the optimal value function V^* for the above Chain environment and report the optimal value $V^*(s)$ for each state s. Include a snippet of your code in your LaTeX submission.

(c) Using your solution to part (b) above, find the optimal state-action value function Q^* (you may like to write a small piece of code to compute this from V^*); report $Q^*(s,a)$ for each state-action pair (s,a). Use this to find an optimal deterministic policy π^* and report $\pi^*(s)$ for each state s.

5. (25 points) **Regret Transfer Bound for Logistic Loss.** Consider a binary classification problem with instance space $\mathcal{X}$ and label space $\mathcal{Y} = \{\pm 1\}$, and underlying probability distribution D on $\mathcal{X} \times \mathcal{Y}$. Recall the logistic loss $\ell_{\log} : \{\pm 1\} \times \mathbb{R} \to \mathbb{R}_+$ is defined by

$$\ell_{\log}(y, u) = \ln(1 + e^{-yu}) .$$

For any real-valued function $f : \mathcal{X} \to \mathbb{R}$, denote by $\mathrm{er}_D^{\log}[f]$ the logistic error (expected logistic loss) of f on examples drawn from D: $\mathrm{er}_D^{\log}[f] = \mathbf{E}_{(X,Y)\sim D}\big[\ell_{\log}(Y, f(X))\big]$; denote by $\mathrm{er}_D^{\log,*}$ the lowest possible logistic error over all real-valued functions:

$$\mathrm{er}_D^{\log,*} = \inf_{f : \mathcal{X} \to \mathbb{R}} \mathrm{er}_D^{\log}[f] .$$

Let $\widehat{f} : \mathcal{X} \to \mathbb{R}$ denote any real-valued function, and let $\widehat{h} : \mathcal{X} \to \{\pm 1\}$ be the corresponding binary classifier defined as $\widehat{h}(x) = \mathrm{sign}(\widehat{f}(x))$. In this problem, you will establish the following regret transfer bound for the logistic loss:

$$\mathrm{er}_D^{0\text{-}1}[\widehat{h}] - \mathrm{er}_D^{0\text{-}1,*} \;\leq\; \sqrt{2\Big(\mathrm{er}_D^{\log}[\widehat{f}] - \mathrm{er}_D^{\log,*}\Big)}$$

For each $x \in \mathcal{X}$, let $\eta(x) = \mathbf{P}(Y = +1 | X = x)$ under D, and let $\widehat{\eta}(x) = \frac{1}{1+e^{-\widehat{f}(x)}}$.

(a) Show that $\mathrm{er}_D^{\log,*}$ is achieved by the function $f^* : \mathcal{X} \to \mathbb{R}$ defined by

$$f^*(x) = \ln\left(\frac{\eta(x)}{1 - \eta(x)}\right) .$$

(*Hint: Write* $\mathrm{er}_D^{\log}[f]$ *as* $\mathbf{E}_X[\mathbf{E}_{Y|X}[\ell_{\log}(Y, f(X))]]$ *and consider finding the optimal value* $f^*(x)$ *for each* x *separately.*)

(b) Show that

$$\mathrm{er}_D^{\log}[\widehat{f}] - \mathrm{er}_D^{\log,*} \;=\; \mathbf{E}_X\big[\mathrm{KL}\big(\eta(X) \| \widehat{\eta}(X)\big)\big] .$$

(c) Show that

$$\mathrm{er}_D^{0\text{-}1}[\widehat{h}] - \mathrm{er}_D^{0\text{-}1,*} \;\leq\; 2\,\mathbf{E}_X\left[|\widehat{\eta}(X) - \eta(X)|\right].$$

(*Hint: Use the fact that* $\widehat{h}(x) = \mathrm{sign}(\widehat{\eta}(x) - \frac{1}{2})$. *You should not need to use the specific form of* $\widehat{\eta}$*; this inequality holds for any plug-in classifier* $\widehat{h}$ *obtained from a class probability estimate* $\widehat{\eta}$*.*)

(d) Put the results of parts (b) and (c) together to show the desired regret transfer bound. You will find it helpful to use the following results:

i. For each $x \in \mathcal{X}$:

$|\widehat{\eta}(x) - \eta(x)| \leq \sqrt{\frac{1}{2}\mathrm{KL}\big(\eta(x)\|\widehat{\eta}(x)\big)}$ (by Pinsker's inequality)

ii. For any random variable Z:

$\mathbf{E}[\sqrt{Z}] \leq \sqrt{\mathbf{E}[Z]}$ (by Jensen's inequality, since $g(z) = \sqrt{z}$ is a concave function)